Performing T-tets

```
% load the data
fibre_measure = readtable("C:\Users\pc user\Documents\SEM Images.csv")
```

fibre_measure = 30×2 table

	IMAGE_2	IMAGE_1
1	3.6030	3.3820
2	3.3230	3.3120
3	3.4130	3.3820
4	3.5840	3.2740
5	3.3430	3.1430
6	3.5520	0.9640
7	3.3020	3.2810
8	3.3140	3.3560
9	3.3230	3.1500
10	2.6990	1.8070
11	3.3320	3.3470
12	3.1550	3.3820
13	3.0790	3.3560
14	3.0470	3.1830
15	2.9180	3.1830
16	3.3140	3.2670
17	3.3140	3.3190
18	3.1950	3.2470
19	3.1550	3.7190
20	2.9360	3.6290
21	2.9950	3.0760
22	1.1610	3.0680
23	2.0920	3.5650
24	3.2210	3.2810
25	2.7520	3.6780
26	3.1380	3.4770
27	2.8970	3.4920
28	3.0770	3.0910
29	3.1060	3.3190

	IMAGE_2	IMAGE_1
30	3.2090	3.0590

```
% Let's check for missing values
Missing_values = sum(ismissing(fibre_measure))
```

```
Missing_values = 1×2
     0     0
```

```
% let's check for outliers
Image1 = sum(isoutlier(fibre_measure.IMAGE_1))
```

```
Image1 = 2
```

```
Image2 = sum(isoutlier(fibre_measure.IMAGE_2))
```

```
Image2 = 2
```

```
% Let's fill the outliers again
diameter1 = filloutliers(fibre_measure.IMAGE_1,"clip");
diameter2 = filloutliers(fibre_measure.IMAGE_2,"clip");
fibre_measure.IMAGE_2 = diameter2;
fibre_measure.IMAGE_1 = diameter1;
outliers = [sum(isoutlier(fibre_measure.IMAGE_2)),
sum(isoutlier(fibre_measure.IMAGE_1))]
```

```
outliers = 1×2
     0     0
```

```
% Let's check whether the two variables have normal/gaussian distribution
% for Image 2
[h, p] = adtest(fibre_measure.IMAGE_2);
if h == 0
    disp('Image_2 diameters follows normal distribution')
else
    disp('The diameters do not have normal distribution')
end
```

```
Image_2 diameters follows normal distribution
```

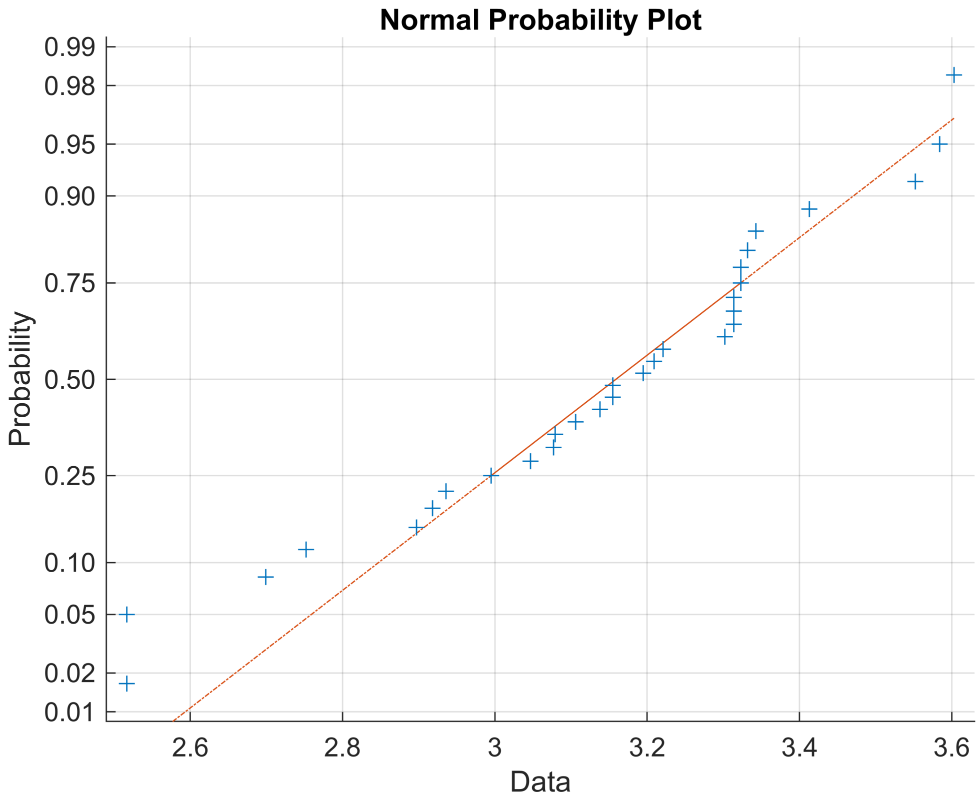
```

% for Image 1
[h, p] = adtest(fibre_measure.IMAGE_1);
if h == 0
    disp('Image_1 diamaters follows normal distribution')
else
    disp('Image_1 diameters have no normal distribution')
end
```

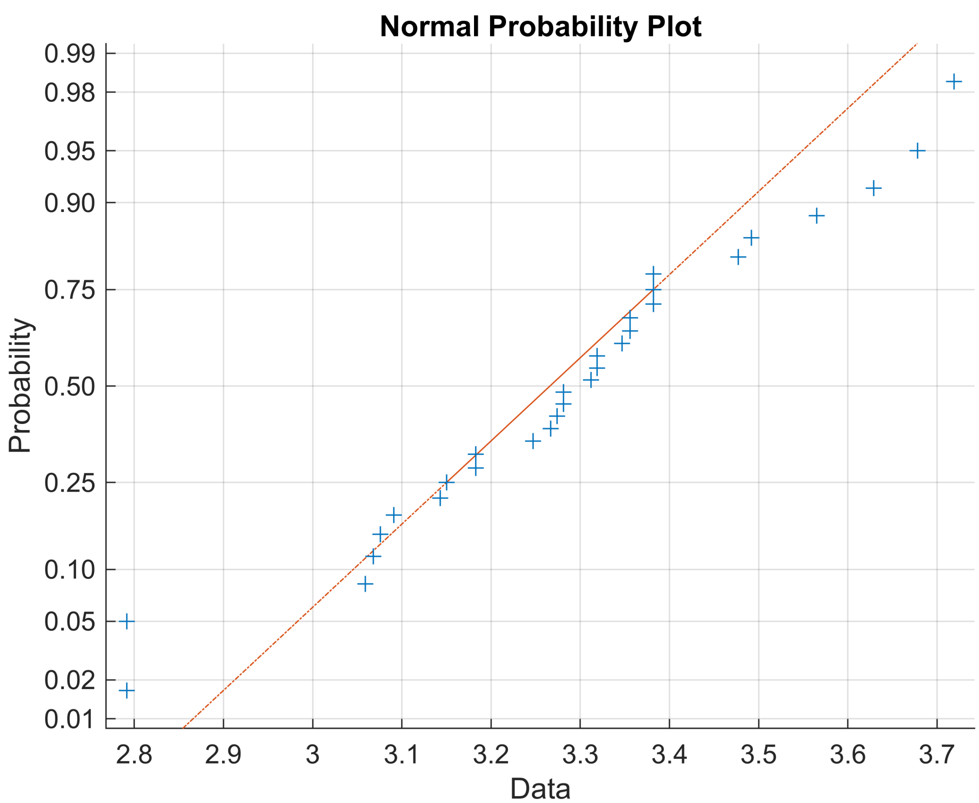
```
Image_1 diamaters follows normal distribution
```

```
% alternatively we can plot the normal curve for each variable
```

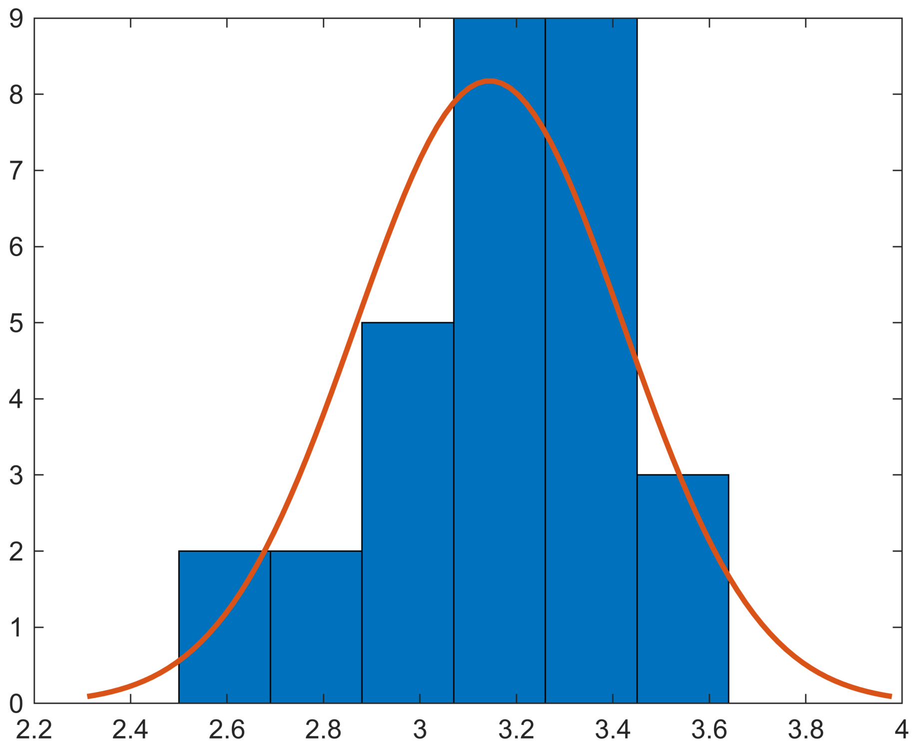
```
normplot(fibre_measure.IMAGE_2)
```



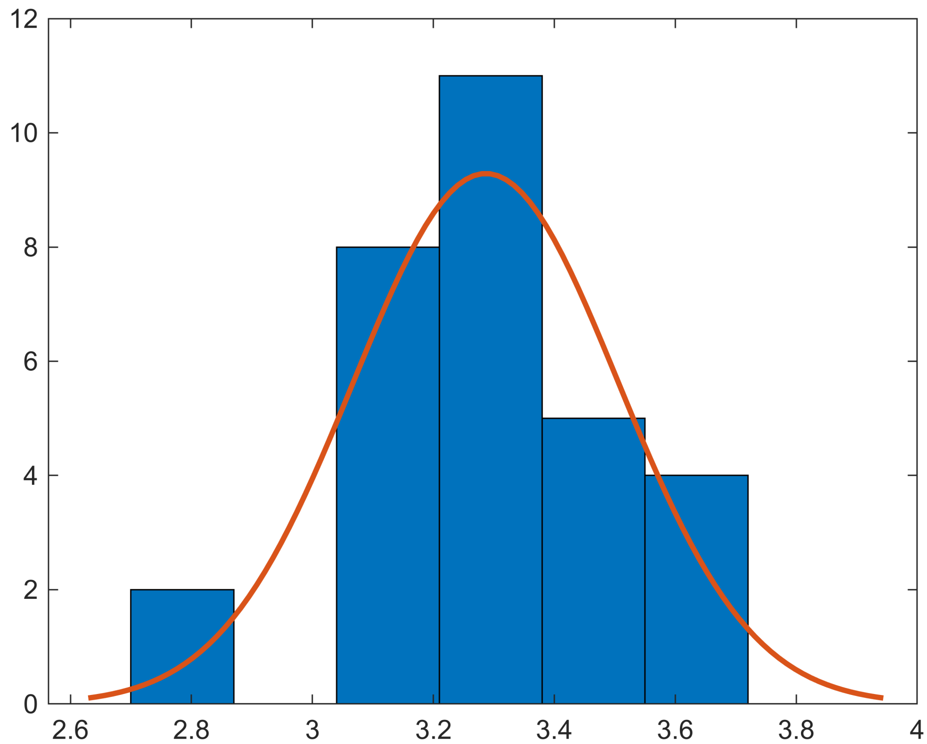

```
normplot(fibre_measure.IMAGE_1)
```



```
% histogram with a curve
histfit(fibre_measure.IMAGE_2)
```




```
histfit(fibre_measure.IMAGE_1)
```




```
% Now let's perform an independent t-test
[h, p, ci, stats] = ttest2(fibre_measure.IMAGE_2, fibre_measure.IMAGE_1);

% Display results
disp(['Hypothesis test results (h): ', num2str(h)]);
```

```
Hypothesis test results (h): 1
```

```
disp(['p-values: ', num2str(p)]);
```

```
p-values: 0.031568
```

```
disp(['Confidence interval: [', num2str(ci(1)), ', ', num2str(ci(2)), ']']);
```

```
Confidence interval: [-0.27177, -0.013021]
```

```
Test_statistics =  struct2table(stats)
```

Test_statistics = 1×3 table

	tstat	df	sd
1	-2.2032	58	0.2503

Visualizing the results

```
% Let's create box plots
figure;
```

```
boxplot([fibre_measure.IMAGE_2, fibre_measure.IMAGE_1], 'Labels', {'Image 2',
'Image 1'});
title('Independent t-test Results for SEM PCL/PLGA fibres');
ylabel('Fibre Diameter in mm');

% Calculate the maximum value for the line
max_value = max([max(fibre_measure.IMAGE_2), max(fibre_measure.IMAGE_1)]) + 0.3;

% Add p-value annotation
text(1.5, max_value + 0.1, sprintf('p = %.3f', p), 'HorizontalAlignment', 'center');

% Add a line indicating the comparison
hold on;
plot([1, 2], [max_value, max_value], 'k-', 'LineWidth', 1.5);

% Adjust y-axis limits to ensure the line is visible
ylim([min([min(fibre_measure.IMAGE_2), min(fibre_measure.IMAGE_1)]) - 0.5,
max_value + 0.5]);

hold off;
```